

Diseño e Implementación de una Aplicación CTI con Georreferenciación de Números Telefónicos Fijos Utilizando el TAPI de la Central Telefónica IP Office de Avaya

Diego Chacón Herrera¹

Pablo Hidalgo Lascano¹

¹Escuela Politécnica Nacional
E-mail: contact@chacondiego.com
E-mail: pablo.hidalgo@epn.edu.ec

Resumen—En este documento se describe el ciclo de vida de creación de un sistema de control de líneas telefónicas de *call center* que incluye funcionalidades de georreferenciación de números telefónicos fijos mediante el empleo de interfaces de programación basados en *Computer Telephony Integration* (CTI) para la central telefónica Avaya IP Office 500 V2.

En primer lugar, se realiza un estudio de las diferentes tecnologías CTI disponibles así como también se describen las características y herramientas de la central telefónica utilizada y sus requisitos para la elaboración de aplicaciones CTI.

Posteriormente, utilizando como metodología de desarrollo Proceso Unificado, se describe el análisis y diseño del sistema (los cuales están relacionados con las fases de inicio y elaboración) en las que se identifican y evalúan los requisitos técnicos, funcionales y opcionales para después determinar los casos de uso, diagramas de flujo y escenarios de interacción entre los distintos elementos que componen la solución, dando como resultado una aplicación distribuida multiplataforma.

El desarrollo del sistema corresponde a la fase de construcción en la que se detallan los elementos, páginas, archivos y librerías finales producidos. En la fase de transición, en cambio, se describe el despliegue de la solución en el escenario de pruebas, así como también se evalúa el grado de cumplimiento de los requisitos primarios definidos en la fase de inicio.

Por último, se realiza un análisis comparativo técnico-funcional y otro de tipo económico con respecto a otras soluciones disponibles en el mercado nacional (una comercial y otra gratuita) para determinar las características clave que hacen de la solución propuesta una alternativa viable para pequeñas y medianas empresas a un costo asequible.

Términos de Indización— *call center*, central telefónica, georreferenciación, ingeniería de software, interfaces de programación de aplicaciones, java, programación, red de área local, sistema cliente-servidor, TAPI, telefonía.

Este proyecto fue parcialmente patrocinado por Lynxsource. Cía. Ltda.
D. Chacón está en el área de Seguridad Informática, Tata Consultancy Services, Quito, Ecuador (e-mail: contact@chacondiego.com).

P. Hidalgo está en el departamento de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador (e-mail: pablo.hidalgo@epn.edu.ec).

I. INTRODUCCIÓN

DE manera general, los sistemas tradicionales de telefonía pura y las aplicaciones institucionales dentro de las empresas operan de forma separada, lo que significa que en la realización o recepción de las llamadas telefónicas no es posible obtener información relevante de los contactos que participan en el proceso. Lo anterior repercute en mayores tiempos de respuesta por parte de los usuarios del servicio de telefonía en el *Call Center* (llámense agentes) y la utilización de dos (o posiblemente más) aplicaciones que proporcionen los datos necesarios para brindar una adecuada atención a los clientes.

En vista que la plataforma de comunicaciones Avaya IP Office no es la excepción, ésta proporciona a sus clientes un conjunto de herramientas que permiten la creación de aplicaciones CTI conforme a las necesidades específicas de cada uno de ellos. Con base en lo anterior, varias soluciones de gestión y control de *call centers*, tanto gratuitas como de pago han sido creadas, la mayoría de ellas de propósito general o con pocas capacidades de adaptación a entornos más específicos.

El anterior es el caso para las instituciones nacionales en las que se requiera identificar rápidamente la localización de una llamada telefónica, como por ejemplo en servicios de salud, emergencias, bomberos o policía, así como también en servicios comerciales como de entrega a domicilio, correos, o taxis. Si fuera posible ubicar el origen de una llamada tan pronto como es contestada, sería posible ofrecer un mejorado nivel de servicio y una atención más rápida, con la ventaja de emplear para ello un solo aplicativo y con ello permitir que los agentes enfoquen sus esfuerzos en tareas directamente relacionadas con el servicio.

Considerando que las soluciones propietarias de gestión de *call centers* disponibles en el mercado para la central telefónica en cuestión no ofrecen la flexibilidad de ajuste al entorno ecuatoriano ni permiten localizar geográficamente las llamadas telefónicas, se ha considerado como valiosa la

elaboración de un sistema que integre los eventos relacionados con las llamadas telefónicas y la información de los contactos almacenada en bases de datos de forma simple y flexible, personalizada para el Ecuador y con la posibilidad de ser actualizada de acuerdo a las variaciones de los planes nacionales de numeración. Dicho sistema, además, ofrece la posibilidad de ser integrada con las bases de datos más utilizadas, así como también de ser desplegada en sistemas operativos propietarios y abiertos. Si a esto se le añade la capacidad de georreferenciar llamadas de números telefónicos fijos y el ofrecer reportes dirigidos al ambiente nacional, se obtiene una solución de gran utilidad, especialmente conveniente para pequeñas y medianas empresas.

II. LA TECNOLOGÍA CTI

Los sistemas telefónicos tradicionales y las tecnologías de la información nacieron de necesidades distintas, en períodos separados. La idea de combinarlos con el objeto de proporcionar un valor añadido a la experiencia del usuario frente a la plataforma de comunicaciones, recibe el nombre de Integración Telefónica por Computador o CTI (*Computer Telephony Integration* por sus siglas del idioma inglés). En otras palabras, CTI es la tecnología que permite la interacción entre teléfonos y computadores de forma recíproca y coordinada.

A. Arquitecturas de Integración CTI

La creación de programas CTI puede seguir cualquiera de las dos arquitecturas siguientes, de acuerdo a las necesidades de las instituciones interesadas.

1) *First-Party (Acceso Directo)*

En este tipo de arquitectura un solo computador controla y manipula las funcionalidades de un teléfono particular, es decir, gestiona los eventos y las llamadas de una sola extensión. Las aplicaciones *first-party* en su mayoría se comunican directamente con el dispositivo telefónico, sin necesidad de utilizar los recursos de la central telefónica o de un servidor de comunicaciones para el efecto, aunque para ello suele ser necesaria la conexión directa entre el computador y el teléfono, pudiéndose requerir tarjetas o conectores adicionales.

2) *Third-Party (Acceso Indirecto)*

La integración *third-party* permite monitorear, administrar y controlar una o más extensiones desde una sola aplicación CTI. Debido a que es posible gestionar otras líneas desde un sistema centralizado, esta alternativa suele ser preferida en escenarios donde se requiere administración avanzada (monitoreo, tarificación, reportes) o donde no es posible instalar o ejecutar aplicaciones *first-party* en los computadores de los usuarios.

B. Estándares CTI

De forma general, la tecnología CTI está basada en Interfaces de Programación de Aplicaciones (o APIs por sus

siglas en inglés), que son un conjunto de procedimientos y métodos publicados como una capa de abstracción (interfaz) para la comunicación entre componentes de software. Entre las más destacadas están:

1) *Interfaz de Programación de Aplicaciones de Telefonía (TAPI)*

Fue desarrollada por Intel y Microsoft en el año 1994 como alternativa a soluciones CTI dependientes del hardware. Es la interfaz más popular debido a su amplio conjunto de funcionalidades y la posibilidad de ejecutar aplicaciones CTI desde cualquier sistema operativo Windows. Define cuatro interfaces: TAPI 2.x (basada en C++), TAPI 3.x (puede ser usada desde C# o Visual Basic), TSPI (Interfaz de Provisión de Servicios de Telefonía) y MSPI (Interfaz de Provisión de Servicios Multimedia). Un TAPI DLL (*Dynamic Link Library*), junto al servidor TAPI (*Tapisvr.exe*) constituyen los componentes principales capaces de abstraer y separar los proveedores del servicio de las aplicaciones CTI. El servidor TAPI rastrea los recursos telefónicos locales y remotos con los TSPs (a través del TSPI). Las comunicaciones que ocurren entre un TSP y un MSP toman lugar utilizando una conexión virtual que atraviesa el TAPI DLL y TAPISRV.

2) *Interfaz de Programación de Aplicaciones de Servicios de Telefonía (TSAPI)*

Desarrollada por Novell y AT&T como una de las primeras interfaces, fue diseñada en un modelo basado en servidor (de tipo *third-party*). Fue ampliamente requerido en ambientes donde se disponía de un gran número de teléfonos analógicos, lo que representaba un menor costo de implementación. Otra de sus ventajas es que era soportado por múltiples sistemas operativos (a diferencia de TAPI que sólo opera en sistemas Windows). Sin embargo, la cada vez creciente popularidad de TAPI y la decisión de Novell de discontinuar el soporte de la interfaz, provocaron su debilitamiento en el mercado informático.

3) *Interfaz de Programación de Aplicaciones de Telefonía de Java (JTAPI)*

El API de Telefonía de Java es una especificación que soporta el control de llamadas telefónicas, diseñada para trabajar tanto en aplicaciones enfocadas a usuario (*first-party*) como en sistemas distribuidos (*third-party*). La última especificación fue publicada en el año 2002 correspondiente a la versión 1.4, y se espera que salga la versión 2.0 que está en desarrollo.

4) *Aplicaciones de Telecomunicaciones con Soporte de Computador (CSTA)*

CSTA, acrónimo del inglés *Computer-Supported Telecommunications Applications*, defendida inicialmente por la empresa Siemens y posteriormente adoptada por la ECMA (*European Computer Manufacturers Association*), provee una capa de abstracción a las aplicaciones de telefonía, independientemente de los protocolos subyacentes que participan en la comunicación. Con un enfoque similar al

TAPI 3.x de Microsoft, fue convertido en un estándar ISO por su carácter internacional. En la actualidad se encuentra en su tercera fase, detallado en el documento ECMA-269.

III. AVAYA IP OFFICE 500 V2

La solución de comunicaciones modular IP Office es una PBX (*Private Branch Exchange*) híbrida con capacidades TDM (*Time Division Multiple Access*) e IP que pueden ser utilizadas de forma independiente o simultánea. Esto quiere decir que la central telefónica es capaz de brindar servicios de telefonía fija analógica, digital e IP.

A. Arquitectura de Comunicaciones IP Office

La solución de comunicaciones IP Office está compuesta por unidades de hardware y aplicaciones de software. Las de mayor relevancia en este documento son:

1) Unidad de Control

El chasis de 2U de altura permite montar la estructura en un rack estándar de 19 pulgadas. Posee cuatro ranuras compatibles con una amplia variedad de tarjetas complementarias o tarjetas base que proporcionan conectividad a teléfonos, y con tarjetas secundarias que proporcionan conexión a proveedores de servicios.



Fig. 1 Vista Frontal de la Unidad de Control IP 500 V2 Junto a un Teléfono IP

El sistema utiliza tarjetas digitales SD que almacenan el S.O. y el archivo de licencias de activación de aplicaciones y funcionalidades. Incluye dos puertos Ethernet, ocho puertos de expansión para módulos externos, un puerto de entrada de audio, un puerto RS232 y un control de relé externo.

2) Tarjetas Base y Tarjetas Troncales

La unidad de control IP500 posee cuatro ranuras para la inserción de tarjetas, que pueden ser tarjetas base o tarjetas troncales. Las primeras incluyen un panel frontal y puertos para conexiones a través de cables. Las tarjetas troncales, en cambio, son montadas sobre las tarjetas base con el propósito de proveer comunicaciones y características adicionales (generalmente conexiones troncales).

3) Módulos Externos de Expansión

Los módulos de expansión brindan conectividad adicional a troncales analógicas y a teléfonos analógicos y digitales;

ampliando de esta forma el número de usuarios de la plataforma.

4) Teléfonos

La central telefónica soporta una amplia variedad de teléfonos analógicos, digitales e IP; pudiendo ser éstos tanto fabricante como equipos compatibles de terceros.

B. Avaya IP Office CTI Link

IP Office ofrece a sus clientes la posibilidad de crear aplicaciones CTI mediante la utilización de *IP Office CTI Link*, un producto de *middleware* CTI basado en estándares y su correspondiente Kit de Desarrollo de Software (SDK). Las interfaces y tecnologías soportadas son descritas a continuación:

1) TAPILink Lite

Interfaz que soporta el desarrollo de aplicaciones CTI de tipo *first-party* mediante el uso del TAPI 2.1 ó TAPI 3.0 de Microsoft. *TAPILink Lite* ofrece características y funcionalidades básicas de control y monitoreo de los dispositivos telefónicos, ajustándose perfectamente a ambientes que requieren de aplicaciones sencillas o de pocos recursos. La principal ventaja del uso de esta herramienta es que se incluye en el CD-ROM de usuario de IP Office libre de cargo, lo que permite su utilización sin la adquisición de licencias.

2) TAPILink Pro

Al igual que la anterior utiliza las interfaces TAPI 2.1 y TAPI 3.0 de Microsoft, cuya principal diferencia es la gestión de varias líneas (extensiones) desde una misma aplicación (o servidor) en un modo *third-party*. Adicionalmente, *TAPILink Pro* provee funcionalidades adicionales respecto a la versión gratuita, entre las más relevantes: habilita la posibilidad de monitorear y controlar grupos de agentes, gestión de inicio y cierre de sesión de usuarios, permite el establecimiento y recuperación de desvío de destinos, habilita la configuración de pertenencia a grupos, etc. Para la creación de aplicaciones CTI que utilicen *TAPILink Pro* debe comprarse una llave de licencia de tipo *CTI Link Pro RFA* que puede ser adquirida de forma similar a cualquier otro producto o funcionalidad de IP Office.

3) TAPI-WAV Driver

Provee soporte para el procesamiento de voz basado en software, exclusivamente para el uso con el TAPI 2.1 de Windows, cuya funcionalidad sólo puede ser utilizada conjuntamente con *CTI Link Pro*. Para TAPI 3.0, IP Office soporta el Proveedor de Servicio Multimedia (MSP) que cumple similares propósitos. Para el uso del TAPI-WAV driver es necesaria la adquisición de licencias *TAPI Wave User*, cada una de las cuales habilita 4 puertos para el procesamiento de voz.

4) DevLink Pro

Provee una interfaz de programación complementaria de las

interfaces *TAPILink Lite* y *TAPILink Pro*. Provee flujos de eventos en tiempo real que contienen información de las actividades de los teléfonos y enlaces troncales. Para su utilización es necesaria la licencia *CTI Link Pro*.

5) *IP Office Station Messaging Detail Record (SMDR)*

Provee una interfaz para la obtención de eventos SMDR. Cuando una llamada es completada, se emite un registro separado por comas respecto al transcurso de la misma, cuya información puede ser dirigida a un puerto y dirección IP específicos. La interfaz IP Office SMDR se utiliza en el desarrollo de aplicaciones de contabilidad o tarificación de llamadas en las empresas.

6) *Kit de Desarrollo de Software*

Contiene la documentación de las interfaces *TAPILink Lite*, *TAPILink Pro*, *DevLink Lite* y *DevLink Pro*. También incluye programas pre-compilados basados en TAPI 2.1 y TAPI 3.0 junto a su código fuente, con el objeto de familiarizar a los desarrolladores con las interfaces antes mencionadas.

IV. ANÁLISIS DE REQUERIMIENTOS Y DISEÑO

A. *Metodología de Desarrollo*

El primer paso en la creación del software informático consistió en la selección de una metodología de desarrollo de software alineada con los objetivos del proyecto. En este sentido, fue seleccionada la metodología de Proceso Unificado (PU) sobre otras más recientes ya que ésta se orienta a la creación de sistemas complejos o de largo plazo, además de ser dirigida por casos de uso, estar centrada en la arquitectura y hacer un uso extensivo de UML (*Unified Modeling Language*) en la documentación. El Proceso Unificado se compone de cuatro fases, cada una de las cuales a su vez se compone por disciplinas que especializan las tareas involucradas en la producción del software.

- La fase de Inicio tiene como objetivo modelar los procesos de negocio de las organizaciones y determinar los principales requisitos del sistema, identificando una arquitectura candidata o los lineamientos generales de la solución. En ella también se realiza un análisis e identificación de riesgos, así como una estimación aproximada de tiempo y costos del proyecto.
- La de Elaboración realiza un profundo análisis de los requisitos y componentes del sistema, así como de su arquitectura, apoyándose principalmente en diagramas de caso de uso, diagramas de clase y definición de paquetes. Las principales metas de esta fase son, por tanto, el tratamiento de los factores de riesgo claves y la validación y diseño inicial de la arquitectura del sistema.
- La fase de Construcción es generalmente la etapa de desarrollo que conlleva más tiempo, basada directamente en los lineamientos definidos en la fase anterior. Cada una de las características del sistema es implementada en una serie de iteraciones, al final de las cuales se obtiene una nueva versión ejecutable del producto. En esta fase comúnmente son utilizados los diagramas de secuencia,

actividades, colaboración, estados, etc.

- Finalmente, la fase de Transición tiene por objeto desplegar la solución en el entorno de los usuarios, cuya retroalimentación favorece el refinamiento y la corrección de problemas de funcionamiento o estabilidad del sistema. Adicionalmente, en esta fase se realiza la capacitación de los usuarios y la creación del material de despliegue, administración y uso de la solución.

B. *Fase de Inicio*

Una vez determinada la metodología, una de las primeras tareas consistió en el modelamiento de los casos de uso del *call center*, ya que la comprensión del funcionamiento de los procesos organizacionales de las instituciones influye enormemente en la creación de las soluciones informáticas. En este sentido, se pudo conocer que en la operación, control y administración de un *call center* existen tres roles principales (agente, supervisor y administrador), para los cuales fueron creados diagramas de casos de uso y de flujo que serían tomados en cuenta en las fases posteriores del desarrollo de la solución. Luego fueron establecidos los principales requerimientos del sistema, catalogados de la siguiente forma:

- Requisitos técnico-operativos, donde se estableció que el empleo del TAPI de Avaya IP Office se convertiría en uno de los ejes principales del diseño de la arquitectura de la solución.
- Requisitos funcionales. En ellos fueron determinados las necesidades e intereses de carácter funcional del sistema. A su vez estos requisitos fueron separados en obligatorios (por ejemplo, el control de las llamadas telefónicas, despliegue de información de contactos en una llamada, monitoreo y supervisión de agentes, gestión de reportes o estadísticas, multiplataforma, etc.) y opcionales (integración con aplicaciones externas, fácil distribución, estabilidad y tolerancia a fallos, etc.).
- Requisitos opcionales. Enfocados a mejorar la experiencia percibida por los usuarios y clientes del sistema (sencillez de uso, bajo coste, entre otros).

Después se realizó una categorización de los requisitos del sistema anterior para encontrar los requerimientos más importantes e influyentes en la determinación de la arquitectura de la solución. La Tabla 1 muestra los requisitos primarios resultantes:

| TABLA I REQUISITOS PRIMARIOS | |
|--|----------|
| REQUISITO | CÓDIGO |
| TAPI de IP Office | IR-RT-T1 |
| Gestión y control de llamadas telefónicas | IR-RF-O1 |
| Despliegue de información de los contactos en una llamada | IR-RF-O2 |
| Monitoreo y supervisión de agentes | IR-RF-O6 |
| Gestión de reportes y/o estadísticas | IR-RF-O7 |
| Gestionar usuarios, servicios y comunicaciones | IR-RF-O8 |
| Multiplataforma | IR-RF-O5 |
| Económico | IR-NF-F1 |
| Gestión e ingreso de observaciones de las llamadas | IR-RF-O3 |
| Detalle del número telefónico de contacto en una llamada externa | IR-RF-O4 |

Posteriormente dichos requerimientos fueron analizados minuciosamente, donde fue posible establecer a *TapiLink Pro* como la versión más favorable por las características avanzadas que permiten la elaboración un sistema de tipo *third-party* de control centralizado de múltiples líneas telefónicas. Además se determinó que la recuperación de la información de los contactos de una llamada telefónica implicaría la integración con bases de datos externas mediante un mecanismo de sólo lectura. También se propuso que el sistema de georreferenciación únicamente sería realizado para números telefónicos fijos, que se utilizarían los planes de numeración provistos por la Agencia de Regulación y Control de las Telecomunicaciones (ARCOTEL) para obtener detalles adicionales de los números telefónicos y que el sistema sería multiplataforma (Fig. 2) para agentes, supervisores y administradores.

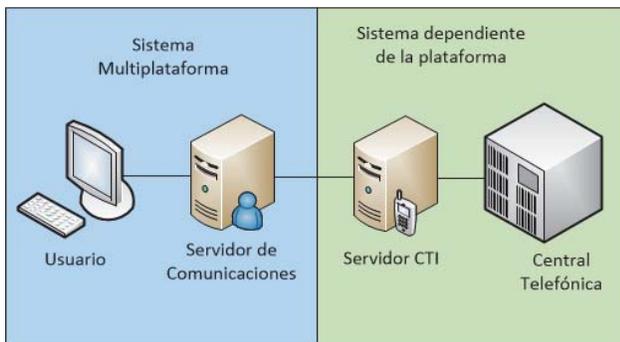


Fig. 2 Sistema Multiplataforma para los Usuarios de la Solución

Habiéndose definido las pautas iniciales del sistema como resultado del análisis de los requisitos primarios, fue posible determinar la arquitectura de la solución, la cual es de tipo cliente-servidor (Fig. 3). En ella, los agentes envían los comandos de gestión y control de llamadas telefónicas a un servidor central, que a su vez procesa y reenvía los comandos necesarios a la central telefónica a través de un servidor CTI intermediario que utiliza TAPI. El servidor principal, además, es el encargado del establecimiento de la conexión con las bases de datos que contienen la información de los clientes que participan en las llamadas telefónicas externas.

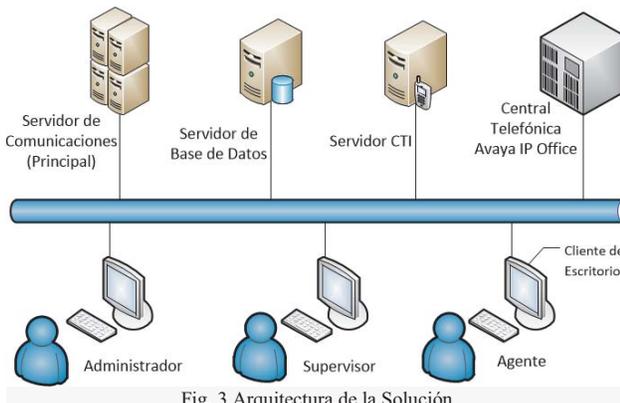


Fig. 3 Arquitectura de la Solución

C. Fase de Elaboración

En primer lugar se profundizó el análisis de los requisitos primarios y secundarios, transformándolos en casos de uso que a su vez fueron refinados con el objeto de establecer a un nivel funcional la interacción que los usuarios tienen con la plataforma (y viceversa). Cuando todos los casos de uso fueron adecuadamente detallados fue posible proseguir con el análisis y diseño mismo de la solución. Es aquí cuando se decidió que el software tomaría como nombre clave **Call Manager**. En la referencia [1] es posible encontrar varias secciones del código fuente de los elementos de la solución y un mayor detalle de su funcionamiento y mutua interacción.

1) Comunicaciones

Dado que los servidores principal (desarrollado en Java) y CTI (escrito en C++) fueron creados en lenguajes de programación diferentes, se consideró apropiado utilizar sockets planos para la interconexión entre los mismos. De igual manera, los agentes se comunican con el servidor principal utilizando la misma tecnología, de tal forma que es posible utilizar un único protocolo de comunicaciones de bajo nivel común para todo el sistema.

2) Inicio y Cierre de Sesión de Agentes

Los eventos de inicio y cierre de sesión tienen una gran importancia ya que son los desencadenadores de la asignación o liberación de los recursos (y por lo tanto, de la gestión de más eventos) que son claves en la operación de los programas de escritorio de los agentes, dando como resultado diagramas de actividades que detallan los pasos a seguir entre los componentes de la plataforma.

3) Gestión de Eventos de las Líneas Telefónicas

En esta sección fue planeado y diseñado el proceso de detección y reenvío de eventos de las líneas telefónicas. Éstos pueden ser generados por la central telefónica (como la detección de una llamada entrante) o por el agente de escritorio (como por ejemplo el deseo de realizar una llamada saliente). En el primer caso, el servidor CTI detecta el cambio en el estado de una línea particular, que es comunicado al servidor principal y es a su vez reenviado al agente de escritorio, mientras simultáneamente se efectúa el proceso de búsqueda de información adicional del contacto en BDD externas así como la búsqueda de coincidencias conforme a los planes de marcación nacionales. En el segundo caso, los eventos producidos por los agentes inicialmente son enviados al servidor principal que, luego de validar los datos de sesión y recursos se encarga de reenviar las peticiones al servidor CTI para que a través de TAPI efectúe las peticiones en la central telefónica.

4) Despliegue de la Información de Contacto en una Llamada

En el transcurso de una llamada, el servidor principal se encarga de realizar la búsqueda de información adicional de los contactos en bases de datos externas mediante el empleo de JDBC (*Java Database Connectivity*). En este caso debe

notarse que los datos de los contactos tienen que haber sido almacenados previamente, los cuales pueden incluir coordenadas geográficas que pueden ser utilizadas para la visualización en un mapa. Se soportaría entonces, la lectura de una única tabla o vista en modo de sólo lectura para recuperar tal información adicional.

5) *Detalle del Número Telefónico Participante en una Llamada Externa*

Son utilizados los planes de numeración publicados por la ARCOTEL para determinar el nombre de la operadora (números fijos y celulares) y la provincia (números fijos solamente) en el transcurso de una llamada. Dado que dichos planes son actualizados constantemente, éstos pueden ser borrados y actualizados en el sistema mediante la carga de archivos de tipo de Valores Separados por Comas (.csv) bajo el estándar RFC 4180 que incluirán los campos: nombre de la central, serie asignada, provincia y operadora.

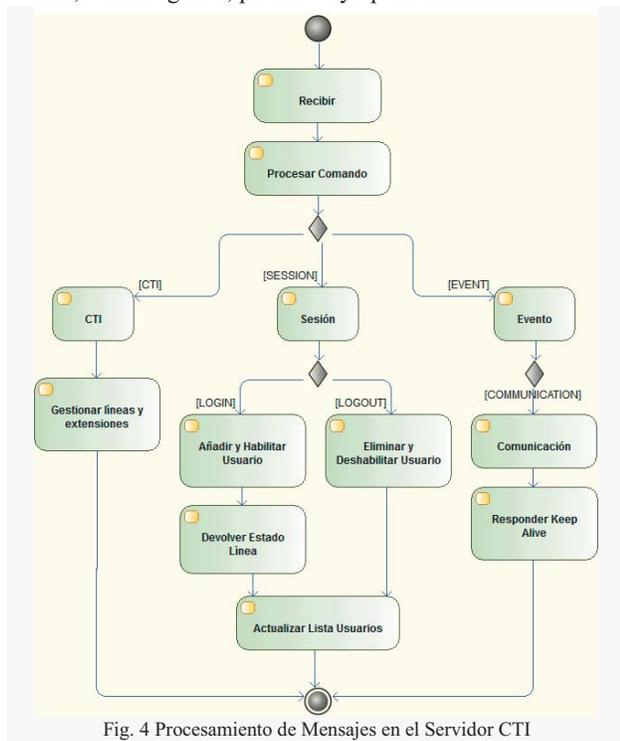


Fig. 4 Procesamiento de Mensajes en el Servidor CTI

6) *Servidor CTI*

Desarrollado en C++, utiliza TAPI 2.x como interfaz para el envío y recepción de eventos desde/hacia la central telefónica. En resumen, el servidor CTI actúa como intermediario entre el sistema informático y el sistema de telefonía. Al estar comunicado con el servidor principal, recibe y procesa distintos tipos de mensajes, como se presenta en la Fig. 4.

7) *Servidor Principal*

Es el componente de mayor complejidad del sistema, punto central de intercambio de información entre los módulos de agente, bases de datos, y aplicativo CTI, así como el encargado de presentar una interfaz de administración y

configuración a supervisores y administradores de la plataforma. Tomando en cuenta la naturaleza multiplataforma y distribuida de la solución, este componente es desarrollado en Java EE (*Enterprise Edition*), incluyendo de esta forma componentes de negocio en los que se procesa de forma centralizada la lógica de negocio de la aplicación que permiten gestionar las comunicaciones con otros componentes del sistema, gestionar las sesiones de los agentes de escritorio, procesar los mensajes y comandos transferidos entre los módulos de agentes y CTI, interactuar con bases de datos, entre otras características. Para ello se utilizan EJBs de tipo *singleton* y de sesión sin estado (*stateless beans*) según las funcionalidades requeridas. Además, el servidor principal incluye un *front-end* de componentes web que proporcionan una interfaz de administración, supervisión y uso de la plataforma desde distintos navegadores, mediante el empleo de la tecnología *JavaServer Faces* (JSF) para la creación dinámica de páginas web. JSF agrupa un diverso grupo de elementos como *beans* administrados, validadores, escuchadores, etc. que interactúan con los componentes de negocio para recuperar y presentar de forma integral información del sistema. Cada rol (agente, supervisor y administrador) de *Call Manager*, por lo tanto, es capaz de acceder a un conjunto de páginas web en el servidor principal.

8) *Aplicativo de Escritorio*

El cliente de escritorio tiene por objeto enviar y recibir eventos en tiempo real que pueden ser generados por el agente (como por ejemplo, el cierre de sesión), por el sistema (se ha perdido la comunicación) o por la línea telefónica (se ha recibido una nueva llamada telefónica). Al ser escrito en Java, utiliza el patrón *Observer/Observable* para separar un hilo dedicado exclusivamente al manejo de la comunicación con el servidor principal del hilo principal de la aplicación. El árbol de procesamiento de mensajes es definido de acuerdo al protocolo común definido en la sección de análisis y diseño de comunicaciones.

9) *Base de Datos*

La BDD local es accedida mediante el empleo del API de Persistencia de Java (JPA), el cual tiene por objeto utilizar objetos Java planos (o POJOs) en la interacción con bases de datos relacionales. Es por ello que ha sido creado un paquete llamado *Entities* que contiene las entidades que mapean las tablas del sistema *Call Manager*. Además un paquete denominado *Facade* alberga *beans* sin estado (*stateless EJBs*) encargados de proporcionar los métodos necesarios para que otros componentes del sistema puedan acceder a la información almacenada, utilizando para las consultas el Lenguaje de Consultas de Persistencia de Java (JPQL). Las bases de datos externas, por otro lado, son accedidas mediante JDBC y requieren de consultas predeterminadas que emplean SQL (*Structured Query Language*). Los gestores de bases de datos externos soportados son PostgreSQL, MySQL y Microsoft SQL Server al ser los más populares.

V. DESARROLLO, DESPLIEGUE Y PRUEBAS

A. Fase de Construcción

En esta fase se crea el producto final que soluciona las necesidades de los usuarios, dividido en cuatro elementos principales.

1) Servidor CTI

Debido a que TAPI sólo puede ser utilizado en sistemas operativos Windows, este componente fue creado utilizando el Entorno de Desarrollo Integrado (IDE) *Microsoft Visual Studio 2008 Professional Edition*. Dicho componente tiene como base los ejemplos proporcionados por Avaya, por lo que ha sido reutilizado el modelo básico de operación y de gestión del TAPI y las líneas telefónicas, con las obligatorias modificaciones y adiciones propias requeridas por el proyecto. Entre las más importantes está la gestión de múltiples líneas telefónicas (funcionamiento de tipo *third-party*) y el empleo de un *mutex* para limitar el número de instancias por servidor a uno. En un mayor detalle, la iniciación de los servicios prestados por este módulo (como por ejemplo la carga del TAPI, la apertura de las líneas telefónicas o la escucha en el puerto de comunicaciones) se realiza en la función *SetupStart* de la clase *CCallManagerDlg*, hilo principal de la aplicación. A continuación se habilita el procesamiento, envío y recepción de eventos CTI con el servidor principal. Finalmente, se actualiza la interfaz con el objeto de presentar los resultados del inicio del aplicativo al administrador. El resultado de la creación de este módulo es un archivo ejecutable llamado *CallManager.exe* que utiliza un archivo de configuración *CallManager.conf*.

2) Biblioteca Común

Fue creada una biblioteca que reúne las clases y enumeraciones comunes utilizadas tanto por el cliente de escritorio como por el servidor principal para evitar redundancias de programación, la cual exporta como resultado un archivo de Java denominado *CallManagerCommons.jar*.

3) Servidor Principal (CORE)

Para su implementación se creó un proyecto de tipo Aplicación Empresarial (*Enterprise Application*) de Java EE 7 en *NetBeans 7.3* que tiene por nombre *CMWI (Call Manager Web 1)*. Para la ejecución de este componente se utiliza el servidor de aplicaciones *GlassFish Open Source Edition*. La elaboración de este servidor produce dos módulos principales, *CMWI-ear*, que es donde se procesa la lógica de negocio de la aplicación y *CMWI-war*, que es donde se crean las páginas de web de administración de la solución. Con respecto a este último, se seleccionó a *PrimeFaces* como *framework JSF* para la creación de páginas web que permita mejorar la experiencia del usuario final y la calidad del producto. La tecnología de acceso a datos, por otro lado, emplea *EclipseLink* para recuperar información desde el gestor de base de datos local *PostgreSQL*.

Adicionalmente, fueron incluidas medidas de tolerancia a fallos para gestionar la reconexión automática entre los

servidores principal y CTI mediante la transmisión periódica de mensajes de tipo *keepalive* que comprueben periódicamente el estado de la comunicación. Por otra parte, para cuando no se pudiera encontrar coordenadas geográficas en el repositorio de información externo (BDD), se utilizó como alternativa el servicio de la CNT de disponibilidad geográfica (ver enlace: <http://gis.cnt.com.ec/giscnt/php/dispoapi.php>). Para que eso fuera posible, se empleó la biblioteca *Apache HTTPComponents* desde uno de los EJBs de procesamiento de eventos.

En cuanto a la implementación de los componentes web del servidor principal, fueron creadas varias páginas web conforme a los roles del *call center*, entre las más importantes:

- Inicio de sesión (para todos los usuarios).
- Administración. La página principal (Fig. 5) presenta el número de agentes conectados y el estado de la comunicación con el servidor principal. Otras páginas permiten la creación y configuración de puestos de trabajo (líneas telefónicas), usuarios del sistema, parámetros de comunicación, planes de numeración, variables del sistema, interconexión con BDD externas y cierre de sesión.

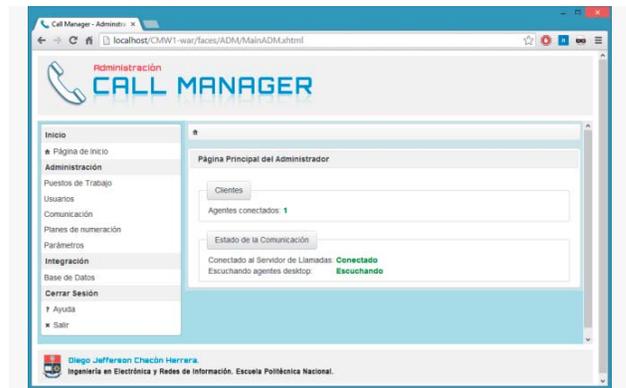


Fig. 5 Página Principal de Administración

- Supervisión. La página principal del supervisor (Fig. 6) muestra el estado de las líneas telefónicas de los agentes conectados en tiempo real. La sección de reportes/estadísticas contiene páginas que muestran el desempeño de los agentes, las llamadas procesadas en el *call center*, llamadas por operadora, y zonas de calor por provincias.

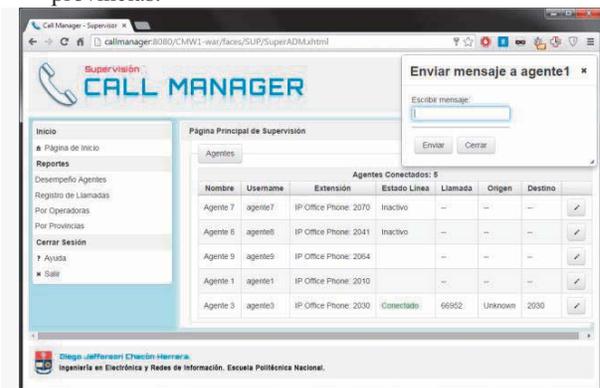


Fig. 6 Página Principal de Supervisión

- Agente. La página principal del agente (Fig. 7) le permite buscar los registros de las llamadas procesadas por el sistema. Otras páginas incluyen la actualización del perfil, la posibilidad de descargar el aplicativo de escritorio desde el servidor principal vía web, y el cierre de sesión.

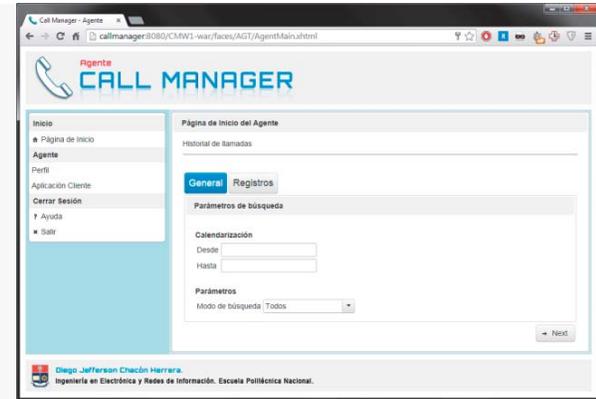


Fig. 7 Página Principal del Agente

Como resultado de la implementación del servidor principal fue obtenido un archivo de extensión *.ear* (*Enterprise ARchive*) de nombre *CMW1.ear* que puede ser desplegado en el servidor de aplicaciones *GlassFish* para su instalación.

4) Cliente de Escritorio

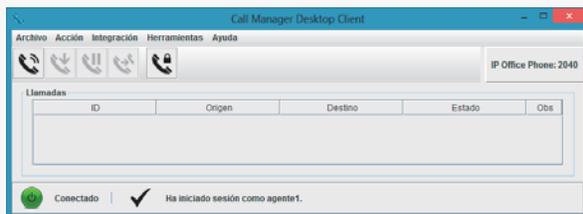


Fig. 8 Ventana del Aplicativo de Escritorio en Estado Inactivo

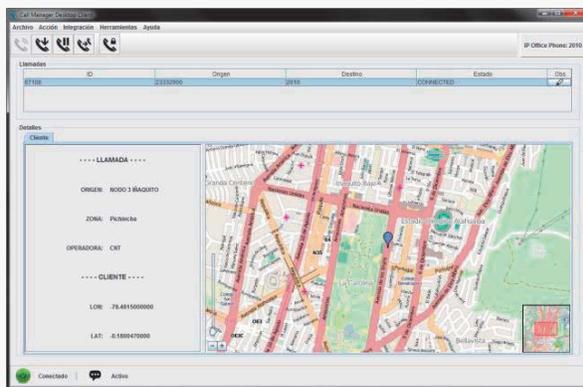


Fig. 9. Aplicativo de Escritorio en el Transcurso de una Llamada

Este último componente, de mayor interacción con el usuario final (Figs. 8 y 9), fue desarrollado en Java SE (*Standard Edition*) de forma reiterativa y simultánea conforme al desarrollo del servidor principal. El motor de generación y presentación de mapas que es utilizado para la georreferenciación de llamadas es *OpenStreetMaps*, el cual fue elegido sobre alternativas como *Bing Maps*, *Google Maps* o *MapQuest* debido a que no requiere licenciamiento adicional

para versiones de escritorio. Para su empleo se ha utilizado la librería de java *JXMapView* que permite incrustar servicios de mapeo en aplicaciones basadas en *Swing* (API de provisión de interfaces gráficas de usuario), siendo compatible con *OpenStreetMaps*. El resultado final de la creación de este módulo es el archivo Java *CallManagerDesktopClient.jar* compatible con el JRE 1.7 (*Java Runtime Environment*) o superior que requiere de las siguientes bibliotecas externas para su funcionamiento:

- *CallManagerCommons.jar*.
- *javaws.jar* (para que pueda ser distribuido desde el navegador web).
- *swing-worker.jar*, *swingx.jar*, *swingx-bean.jar*, *swingx-2007_10_14.jar* (bibliotecas del conjunto *JXMapKit* utilizados para la georreferenciación de llamadas).

B. Fase de Transición

En esta última fase fueron realizadas varias actividades. Una de ellas fue la gestión del despliegue de los componentes (creación de instaladores y requisitos de instalación) para los elementos que así lo demandaron. Otra fue la verificación de cumplimiento del proyecto conforme a las necesidades de los usuarios mediante la realización de pruebas de aceptación. Además en esta etapa se recibió la retroalimentación de los usuarios finales con el propósito de corregir cualquier fallo no previsto y finalmente fue creada la documentación orientada al cliente como manuales y guías de administración y uso.

En lo concerniente a las pruebas, el escenario de evaluación de la solución tuvo lugar en el Laboratorio de Redes II del edificio de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, en la ciudad de Quito. La red para la ejecución de las pruebas fue organizada en subredes separadas en VLANs para datos, voz, y servidores (Fig. 10). La central telefónica Avaya IP Office 500 V2 fue configurada por el autor D. Chacón conforme a las recomendaciones y lineamientos del fabricante, incluyendo el licenciamiento apropiado conforme a los requisitos del sistema. Además, este equipo se interconectó a la PSTN mediante una troncal SIP que apuntaba a la central telefónica interna marca SIEMENS de la EPN.

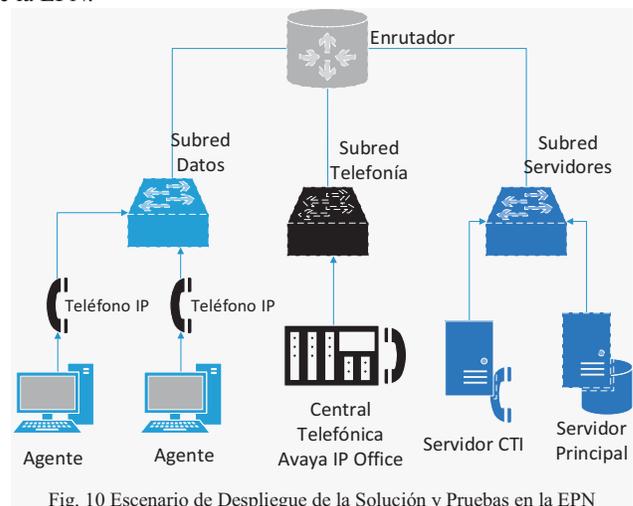


Fig. 10 Escenario de Despliegue de la Solución y Pruebas en la EPN

Específicamente, las pruebas se ejecutaron con 5 agentes, cada uno de los cuales disponía de un computador y su correspondiente teléfono IP Avaya 1608i. Entre las pruebas se recibieron llamadas entrantes (desde la PSTN e internas) y se realizaron llamadas salientes (hacia la PSTN y hacia otras extensiones). En el caso de tratarse de números telefónicos fijos, se comprobó el funcionamiento del agente de escritorio y el correspondiente despliegue de la georreferenciación de las llamadas. Tanto el servidor principal como los computadores de los agentes accedían Internet para obtener las coordenadas y mapas geográficos de los números telefónicos fijos.

TABLA II
CUMPLIMIENTO GENERAL DE REQUISITOS PRIMARIOS

| Código Prueba | Requisito Primario | Verificación |
|---------------|--|--------------|
| TP-CR-P1 | Utilización de TAPI de IP Office | Cumplido |
| TP-CR-P2 | Gestión y control de llamadas telefónicas | Cumplido |
| TP-CR-P3 | Despliegue de información de los contactos en una llamada | Parcial |
| TP-CR-P4 | Monitoreo y supervisión de agentes | Cumplido |
| TP-CR-P5 | Gestión de reportes y/o estadísticas | Cumplido |
| TP-CR-P6 | Gestionar usuarios, servicios y comunicaciones | Cumplido |
| TP-CR-P7 | Multiplataforma | Parcial |
| TP-CR-P8 | Económico | Cumplido |
| TP-CR-P9 | Gestión e ingreso de observaciones de las llamadas | Cumplido |
| TP-CR-P10 | Detalle del número telefónico de contacto en una llamada externa | Cumplido |

Adicionalmente fueron creadas tres bases de datos de prueba en los gestores *PostgreSQL*, *MySQL Server* y *SQL Server 2008* desde los cuales se recuperó información de números telefónicos fijos predefinidos con el propósito de comprobar la integración con bases de datos externas. Así, el cumplimiento de los requisitos primarios conforme a las pruebas realizadas quedó documentado según la tabla II.

VI. ANÁLISIS COMPARATIVO

El reto al que se enfrenta *Call Manager* en lo que se refiere a una evaluación técnica y económica está determinado por el conjunto de funcionalidades comunes con respecto a otras alternativas comerciales. Luego de una extensiva búsqueda de aplicaciones comerciales de geolocalización de llamadas telefónicas nacionales solamente se pudieron encontrar sistemas propietarios orientados a sistemas de emergencia, especialmente dirigidos a la provincia del Azuay. Esto significa que existe una oportunidad de mercado enorme que puede ser aprovechada por *Call Manager* como el primer intento comercial de localización de llamadas telefónicas para pequeñas y medianas empresas en el Ecuador.

A. Técnico-Funcional

La tabla III provee una lista de características comunes que tienen *Avaya IP Office Contact Center* y *Evolution Call Center* (sistemas de gestión de *call centers*) con la solución creada en este proyecto. Nótese que ninguna de ellas proporciona funcionalidades de georreferenciación de llamadas telefónicas y menos aún soporte específico para el

entorno ecuatoriano. Del mismo modo, *Call Manager* no posee funcionalidades de IVR, campañas de llamadas salientes, o gestión de comunicaciones multicanal, al no haber sido parte del alcance ni de los objetivos del proyecto. La selección de una alternativa sobre otra a nivel técnico, por tanto, deberá estar sujeta a las necesidades específicas de cada institución, incluso pudiendo coexistir *Call Manager* junto a otras soluciones de *call center* en caso de ser necesario sin superponer funcionalidades entre ellas.

TABLA III
CUMPLIMIENTO GENERAL DE REQUISITOS PRIMARIOS

| Característica\Solución | Avaya IP | | |
|--|--------------|-----------------------|-----------------------|
| | Call Manager | Office Contact Center | Evolution Call Center |
| Gestión del estado del teléfono de los agentes | Sí | Sí | Sí |
| Georreferenciación de llamadas telefónicas | Sí | No | No |
| Autenticación de usuarios y perfiles | Sí | Sí | Sí |
| Ingreso de observaciones de las llamadas | Sí | Sí | Parcial |
| Administración vía Web | Sí | Sí | Sí |
| Monitoreo en tiempo real de los agentes | Sí | Sí | Sí |
| Visualización de registros, reportes o estadísticas | Sí | Sí | Sí |
| Envío de mensajes a los agentes | Sí | Sí | No |
| Interconexión con bases de datos externas | Sí | Sí | Sí |
| Soporte de PBXs de otros fabricantes | Parcial | No | Sí |
| Integración con aplicaciones externas | Parcial | Sí | Sí |
| Atractivo y de fácil uso | Sí | Sí | Sí |
| Mapas de calor para llamadas entrantes y salientes | Sí | No | No |
| Compatibilidad de acuerdo a los planes de numeración del Ecuador | Sí | No | No |
| Reporte de llamadas por operadora | Sí | No | No |
| Multiplataforma | Parcial | No | No |
| Respuesta de voz Interactiva | No | Sí | Sí |
| Grabación de llamadas | No | Sí | Sí |
| Campañas de llamadas salientes | No | Sí | Sí |
| Contacto multicanal con los clientes (e-mail, chat web, etc.) | No | Sí | Sí |
| Autoservicio | No | Sí | No |
| Enrutamiento basado en habilidades | No | Sí | Sí |

B. Económico

En el análisis económico fueron tomados en cuenta dos factores importantes: en primer lugar, no se incluye el licenciamiento específico requerido por la central telefónica al ser responsabilidad de las instituciones interesadas. En segundo, no fueron comprendidos los requerimientos de

hardware y software específicos sobre los que operan las soluciones de gestión de software, debido a que las instituciones pueden reutilizar infraestructura existente o adquirirla directamente a proveedores estratégicos. Tomando en cuenta lo anterior, los costos comparativos de despliegue para un *call center* básico de 5 agentes es el siguiente:

1) *Call Manager*

Un enfoque práctico de cálculo mediante costos fijos estableció un precio base para las instituciones de \$2.713,723; habiéndose considerado 8 futuros posibles clientes con un margen de rentabilidad del 15% que incluye servicios de instalación e IVA. Cabe precisar que, con la excepción del servidor CTI que puede ser instalado en cualquier equipo Windows, todo el software relacionado puede ser obtenido de manera gratuita, reduciendo de forma considerable los costos de despliegue de la solución que incluso puede ser virtualizada o agrupada en un único servidor físico. Además es importante notar que no es necesario utilizar teléfonos especiales o diademas para el funcionamiento normal del *call center*.

2) *Avaya IP Office Contact Center*

Al ser una aplicación propietaria, los clientes de este sistema están cubiertos en temas de garantía y soporte. Un despliegue básico de *IP Office Contact Center* tiene un costo de \$8.579,20 dólares (incluye IVA). En este precio se incluyen los costos de los equipos terminales (IP Phone 9608G) y diademas para 5 agentes de *call center* requeridos por la solución. Debe tomarse en cuenta que los sistemas de contact center, grabación y estaciones de trabajo, todos requieren de sistemas operativos Windows 7, 2008 o superior.

3) *Evolution Call Center*

Si bien la versión *Community* se entrega de forma gratuita mediante el llenado de un formulario, no se cuenta con documentación avanzada, garantía, o soporte por parte del proveedor. Adicionalmente, tanto el servidor como los clientes deben contar con el sistema operativo Windows para su operación, así como también son necesarios teléfonos Avaya de la serie 96XX, incurriendo en gastos adicionales. Considerando únicamente los costos de los teléfonos (junto a las diademas y los adaptadores de energía) para 5 agentes el monto inicial es de \$2.005,00 (incluido IVA).

Como resultado, si bien las tres soluciones comparadas poseen ciertas características comunes, las funcionalidades específicas de *Call Manager* hacen que ésta sea una alternativa válida y conveniente para casos concretos en los que la georreferenciación de llamadas sea prioritaria. Además, el precio del producto por institución está a la par (en cuanto a rango de precios se refiere) con alternativas comerciales que le permiten ser asequible para pequeñas y medianas empresas del Ecuador.

VII. CONCLUSIONES

Los resultados obtenidos en la disciplina de pruebas de la fase de transición permitieron concluir que el sistema de

gestión de llamadas de *call centers Call Manager* cumplió positivamente las expectativas y requerimientos planteados en las fases de inicio y elaboración, haciendo de esta solución una alternativa por demás conveniente para los sectores en los que se requiera establecer rápidamente la localización geográfica de las llamadas telefónicas como son: servicios de emergencia, bomberos y policía, transporte (taxis por ejemplo), correo, entrega a domicilio, entre otros.

Tomando en cuenta que en el mercado ecuatoriano para pequeñas y medianas industrias no existen soluciones de propósito general que permitan georreferenciar llamadas telefónicas, *Call Manager* constituye el primer intento que ofrece una solución uniforme y general que se adapte no sólo a *call centers* que emplean la plataforma Avaya IP Office sino también a aquellos en los que operan centrales telefónicas compatibles con TAPI con mínimos cambios.

Uno de los mayores inconvenientes en la elaboración de sistemas distribuidos recae en el aumento del número de posibles puntos de falla que pueden afectar la operación del sistema. Si bien se trató de que el software fuera tolerante a fallos, por razones de tiempo y de recursos no fue posible abarcar todos los escenarios de error que pueden ocurrir en la operación no sólo de *Call Manager* sino también de la central telefónica, bases de datos, o teléfonos y computadores de los agentes. Sin embargo, las pruebas realizadas determinaron que bajo condiciones normales de operación el sistema trabaja de forma estable de acuerdo a lo esperado durante prolongados períodos de tiempo.

Aunque la utilización de la herramienta es sencilla e intuitiva, los procedimientos previos de instalación de los componentes principales pueden parecer complicados debido a la arquitectura propia de la solución. No sólo deben ser analizados planes de numeración o permisos de marcación, sino también tienen que considerarse parámetros de licenciamiento, conectividad de red, configuraciones de BDD, etc. para que *Call Manager* pueda cumplir con los objetivos propuestos.

La naturaleza distribuida del sistema permite añadir módulos y funcionalidades que puedan expandir los alcances iniciales de la solución. Las posibilidades de personalización y crecimiento de la solución son enormes, especialmente si tienen como base el escenario nacional al que está dirigida.

RECONOCIMIENTOS

Un agradecimiento a Lynxsource Cía. Ltda., especialmente al MSc. David González, a la Ing. Johanna Pinargote y al Ing. Franklin Quiroz por el invaluable apoyo, documentación e infraestructura provisto para la creación de este software.

REFERENCIAS

- [1] D. Chacón, and P. Hidalgo, "Diseño e Implementación de una Aplicación CTI con Georreferenciación de Números Telefónicos Fijos Utilizando el TAPI de la Central Telefónica IP Office De Avaya", Facultad Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador, 2015.
- [2] W. Yarberry, *Computer Telephony Integration*, 2nd. ed., Auerbach Publications. USA, 2003.

- [3] R. Walters, *Computer Telephony Integration*, 2nd. Ed., ArtechHouse. Londres, 1999.
- [4] A. Sulkin, *PBX Systems for IP Telephony*, McGraw-Hill. 2004.
- [5] ARCOTEL, *Plan Técnico Fundamental de Numeración (PTFN)*, Resolución TEL-068-04-CONATEL-2013.
- [6] Avaya, *IP Office 9.0.3 Product Description*, Cod. 15-601041 Issue 27.04. 2014.
- [7] Avaya, *Avaya IP Office ACSS Conversion Training*, Código 5S999920.
- [8] Avaya, *Ventas de IP Office*, Cod. ASC00121WEN.
- [9] J. Proise, *Programming Windows with MFC*, Microsoft Press, 1999.
- [10] Oracle, *The Java EE 7 Tutorial*, Cod. E39031-01. Oracle Corporation. 2014.
- [11] J. Schuller, *Aprendiendo UML en 24 Horas*, Prentice Hall.
- [12] R. Pressman, *Ingeniería del Software*, 6ta. ed., Mc-Graw Hill.
- [13] I. Jacobson, G. Booch and J. Rumbaugh, *The Unified Software Development Process*, Pearson Education, 1999.
- [14] M. Bodin and K. Dawson, *The Call Center Dictionary*, CMP Books, 2002.
- [15] D. Sharp, *Call Center Operation: Design, Operation, and Maintenance*, Digital Press, 2003.
- [16] Avaya, *TAPI Link Installation*. Cod. 15-601034. 2013.
- [17] Avaya, *TAPILink Developer's Guide*, Cod. 15-601035. 2013.
- [18] Ç. Çivici, *PrimeFaces User Guide 5.0*, PrimeTek Informatics.
- [19] M. Glinz, *COCOMO (Constructive Cost Model)*, Seminar on Software Cost Estimation, PDF, 2003.
- [20] Avaya, *Avaya IP Office Contact Center Brochure*, 2014
- [21] J. Hidrobo, *Tecnología VoIP y Telefonía IP*, Alfaomega, México, 2006.
- [23] G. Hohpe, and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison Wesley, 2003.
- [23] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*, 3rd. ed., Addison-Wesley, 1994.



Diego Chacón Herrera, Nació en Cayambe el 17 de Octubre de 1987, sus estudios secundarios los realizó en el Instituto Tecnológico Superior Nelson Torres de la ciudad de Cayambe, los estudios universitarios los cursó en la Escuela Politécnica Nacional (EPN) en la carrera de

Ingeniería en Electrónica y Redes de Información.



Pablo Hidalgo Lascano, Director del Proyecto. Ingeniero en Electrónica y Telecomunicaciones de la Escuela Politécnica Nacional. Profesor principal de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional.